REMARKS

In the Office Action dated February 9, 2005, the Examiner rejected claims 1-5, 7-15, and 17-20 under 35 USC 103(a) as unpatentable of Raj (Enterprise JavaBeans) in view of Jain (Java Call Control, Coordination and Transactions, IEEE Communications Magazine, January 2000) and rejected claims 6 and 16 under 35 USC 103 as unpatentable over Raj, Jain, and Shoffner (Write a Session EJB). Claims 1 through 20 remain at issue.

The Art Rejections

The Applicants strongly disagree with the Examiner's rejection. The Examiner has failed to show a prima facie case of obviousness. As noted by the Examiner, Raj fails to teach or suggest a remote interface associated with the entity bean class, the remote interface being arranged to drive the state machine in response to input events. A review of the Jain reference demonstrates that this feature is also not taught or suggested.

The Jain reference is specifically directed to the Java Telephony application programming interface API (JTAPI) which is a portable, object-oriented interface for Java based computing telephony applications. JTAPI is expressed in Java and defines a core call model to support basic call set up, and a number of extensions, mostly designed to model call center features, multi-party conference calls, call routing, etc. See the left column of page 110 of the Jain reference.

As described in the Jain reference, the state of a telephone call is maintained by finite state machines associated with call, connection and terminal connection objects. A complete definition of the various state machines is defined in the Java Telephony API Overview, Version 1.2, October 1997 (JTAPI Specification), published by Sun Microsystems and available at (http://java.sun.com/products/jtapi/jtapi-1.2/Overview.html), a copy of which is listed in the Information Disclosure Statement filed concurrent with this Amendment B.

The finite state machines mentioned in the Jain reference are for calls, connections, etc., respectively. In contrast, the present invention relates to implementing state machines as enterprise beans on an enterprise platform. The state machine is used within a computing environment that supports an enterprise platform that includes an entity bean class, a home

interface associated with the bean class, and a remote interface that is associated with the entity bean class. The entity bean class, the home interface, and the remote interface collectively implement entity objects. Further, with the present invention, the state machine corresponds to an entity object. The home interface is used to create, find, and remove state machines, while the remote interface is used to drive the state machine transitions. The entity objects, entity bean class, the home interface, and the remote interface are realized as an Enterprise JavaBean entity bean component. Accordingly, the Jain reference (and the JTAPI specification) fail to teach or suggest:

> a remote interface associated with the entity bean class, the remote interface being arranged to drive the state machine in response to the input events;

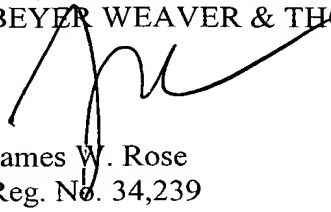
Jain (and the JTAPI specification) fail to teach or suggest the very features that the Examiner is relying in combination with Raj to reject the claims of the present application. Since neither the Raj or Jain references teach or suggest, among other elements of the present invention, a remote interface being arranged to drive the state machine in response to input events, the claims of the present invention are therefore allowable.

Furthermore, Jain is directed to application programming interfaces for Java Telephony. Raj is directed to enterprise JavaBeans. Two references are completely different. There is absolutely no teaching or motivation in either reference for the proposed combination. Furthermore, even if it were proper to combine these references, the combination still would not teach the present invention as claimed. Neither Raj or Jain teach or suggest a remote interface arranged to drive a state machine in response to input events.

The Applicants submit that claim 1 is therefore allowable. Although patentable in their own right, claims 2-10 are allowable based on their dependency on claim 1. Claims 11-20 are also allowable for essentially the same reasons as described above.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



James W. Rose
Reg. No. 34,239

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300